**Table N°1: Registers.**

For every payment done the software creates a record with data about that payment as the amount of money or its date. This table contains all the payment records that may correspond partially or entirely to an inscription, a monthly payment, an advancement, an arrears, a payment concept or a product. All of this concepts own its table where a record is created with the fraction of the payment from a register that correspond to it. The payment of a register may be in transference (in this case indicates the bank it was made from), cash, dollar or a combination of this.

**Properties:**

* **IdRegister.**
* **Date (date):** The date when the payment is done and the register is created.
* **Transfers (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Bank (varchar 45):** Bank’s name the transference was made from if there was an amount of money paid from a bank.
* **Reference (varchar 45):** reference number in transfer.
* **Cash (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Dolars (decimal 8, 2):** the amount of money that was paid with dollar cash.
* **DolarPrice (decimal 20, 2):** conversion rate from BsS to dollar.
* **IdRepresentative:** representative’s id in db that made the payment.

**Table N°2: Balances.**

This table contains the partial amount of money from a register that correspond to either an inscription, monthly payment, advancement or arrear.

**Properties:**

* **IdBalance.**
* **Transfers (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Cash (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Dolars (decimal 8, 2):** the amount of money that was paid with dollar cash.
* **Type (varchar 15):** describe the type of balance, may be either INSCRIPTION, MONTHLYPAYMENT, ADVANCEMENT or ARREAR.
* **IdRegister:** register’s id which correspond the balance.

**Table N°3: paymentsconceptsbalance.**

This table contains the partial amount of money from a register that correspond to a specific payment concept.

**Properties:**

* **IdPaymentConceptBalance.**
* **Transfers (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Cash (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Dolars (decimal 8, 2):** the amount of money that was paid with dollar cash.
* **IdPaymentsConcept:** The payment concept id that correspond to this balance.
* **IdRegister:** register’s id which correspond the balance.

**Table N°4: productsbalance.**

This table contains the partial amount of money from a register that correspond to a specific product.

**Properties:**

* **IdproductBalance.**
* **Transfers (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Cash (decimal 20, 2):** The amount of money in BsS that was paid from this method.
* **Dolars (decimal 8, 2):** the amount of money that was paid with dollar cash.
* **IdProduct:** the product’s id that correspond to this balance.
* **IdRegister:** register’s id which correspond the balance.

**Table N°5: rates.**

This table contains the two (so far) main rates and their prices. They are MONTHLYPAYMENT and INSCRIPTION.

**Properties:**

* **IdRate.**
* **Price (decimal 8, 2):** Price of the rate in dollars.
* **Type: (varchar 20):** Either MONTHLYPAYMENT or INSCRIPTION.

**Table N°6: paymentconcepts.**

Each rate may have one or more payment concepts that are an extra tax of the rate that must be paid.

**Properties.**

* **idPaymentConcept.**
* **Name (varchar 45):** name of the payment concept.
* **Price (decimal 8, 2):** price in dollars of the payment concept.
* **idRate:** rate’s id the payment concept belongs to**.**
* **Deleted:** whether or not the payment should be showed to user, a payment concept may be deleted and not to be available anymore but the register made to this must be still available.

**Table N°7: products.**

This table contains the products that can be sold in a payment. A product may be mandatory for a representative and must buy the product once.

**Properties:**

* **IdProduct.**
* **ProductName (varchar 45):** product’s name.
* **Price (decimal 8, 2):** product’s price in dollars.
* **Mandatory (bool):** whether the product is mandatory or not.
* **Deleted.**

**Table N°8: representative.**

This table contain all the representative of the system and their data.

**Properties:**

* **Idrepresentative.**
* **Names (varchar 45):** representative’s first and second name.
* **lastNames (varchar 45):** representative’s first and second last name.
* **dni (varchar 45):** representative’s dni.
* **Balance (decimal 8, 2):** it is summation of all its students balance.
* **Phone (varchar 45).**
* **Email (varchar 45).**
* **InscriptionPending (double 8, 2):** if the representative complete the inscription process but does not paid it completely the remaining amount is save in this column.
* **idDniType.**
* **Deleted.**

**Table N°9: students.**

Each representative must enroll at least one student, depending on the number of enrolled student the amount of money the representative must pay increase. Each student balance is a fraction of its representative’s balance according with the number of students it represents.

**Properties:**

* **idStudent.**
* **Names (varchar 45).**
* **lastNames (varchar 45).**
* **Dni (varchar 45).**
* **birthDate (date).**
* **Relationship (varchar 45):** the student’s relationship with its representative.
* **Status (varchar 45):** The student’s status may be either REGULAR, SCHOLARSHIP, EXEMPT. This defines whether its representative should pay or not.
* **Blood (varchar 45):** Student’s blood type.
* **Weight (double).**
* **Size (double).**
* **Email (varchar 45).**
* **Phone (varchar 45).**
* **socialMedia (varchar 45).**
* **Balance (decimal 8, 2):** Student’s current balance, it might be positive or negative, it is subtracted for the inscription and every month for the monthly payment price and it is added with every payment.
* **Inscription (bool):** whether or not the student has paid the inscription.
* **paidMonths (int):** Months the students has paid in the current year.
* **idRepresentative.**
* **IdDniType.**
* **idSection.**
* **idGrade.**
* **Deleted.**

**Table N°10: dnitype**

This table contains the possible suffixes of the dnis.

**Properties:**

* **idDniType.**
* **Letter (varchar 1).**

**Table N°11: grades.**

The grades students can enroll in.

**Properties:**

* **idGrade.**
* **scholarYear (varchar 45):** grade’s name.
* **Deleted.**

**Table N°12: sections.**

Each grade must have a least one section and it can have multiple ones. A section is a subdivision of a grade where students enroll in.

**Properties:**

* **idSection.**
* **Section (varchar 45):** Section’s name or letter to identify it.
* **Capacity (int):** the maximum amount of students that can enroll in the section.
* **idGrade:** grade’s id the section belongs to.
* **Deleted.**

**Table N°13: monthpayments.**

When a student make a payment of a month the software creates a record of that month payment, this is for every month, a student may pay for multiple months at one but it will create a record for each one separately. The payment of a month may be partial or complete.

**Properties:**

* **idMonthPayment.**
* **idStudent.**
* **idRepresentative.**
* **Date.**
* **paidMonth (int):** The month that was paid.
* **paidAt (int):** The month when it was paid.
* **paidAmount (decimal 8, 2):** The amount of money that was paid.
* **debthAmounth (decimal 8, 2):** The amount of money that must be paid.
* **Tranfers.**
* **Cash.**
* **Dolars.**
* **scolarYear (varchar 4):** The scolar year when it was paid.

**Table N°14: globals.**

One row table that contains general data to the functionality of the system.

**Properties:**

* **idGlobal.**
* **currentMonth:** current month that must be paid for the students.
* **currentYear:** current scholar year.

**Table N°15: users.**

The list of users that can enter to the system.

**Properties:**

* **idUsers.**
* **Username.**
* **Password.**
* **Question:** Security question in case of password lost.
* **Answer.**